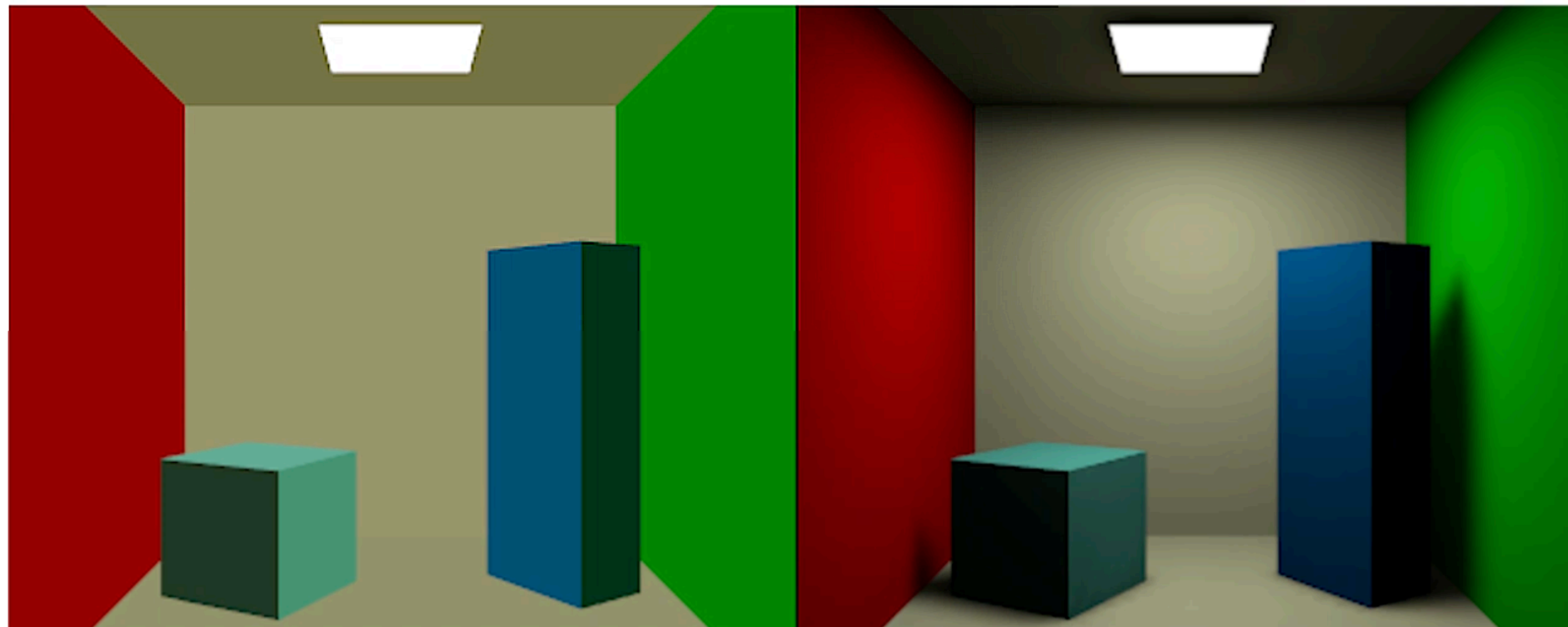


# **ALGORITMO DI RADIOSITY**

**Baggio Tommaso, Giugno 2025**

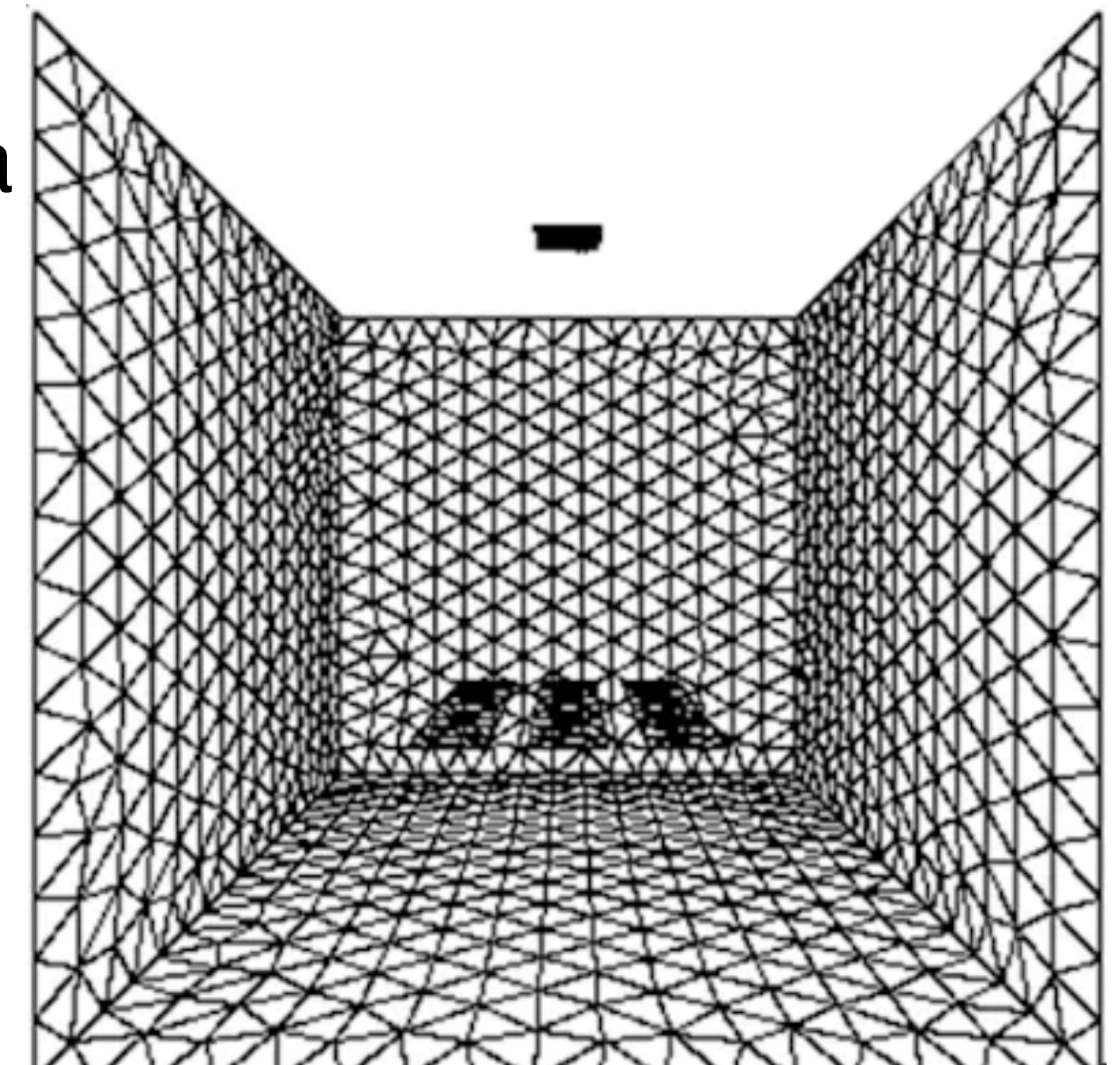
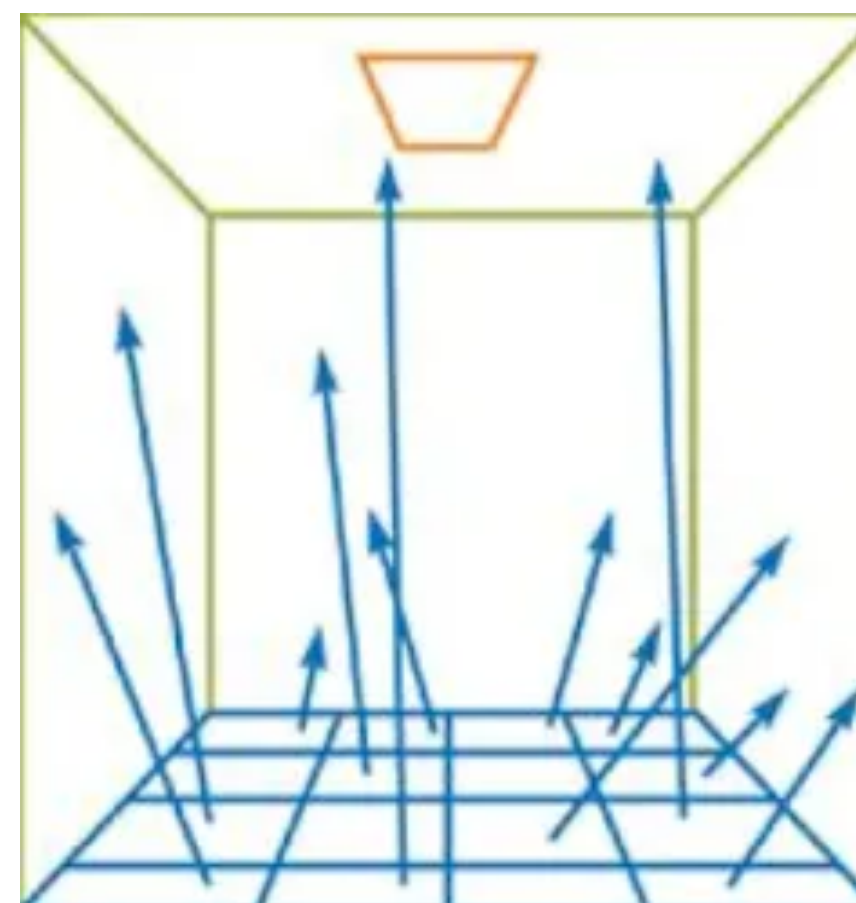
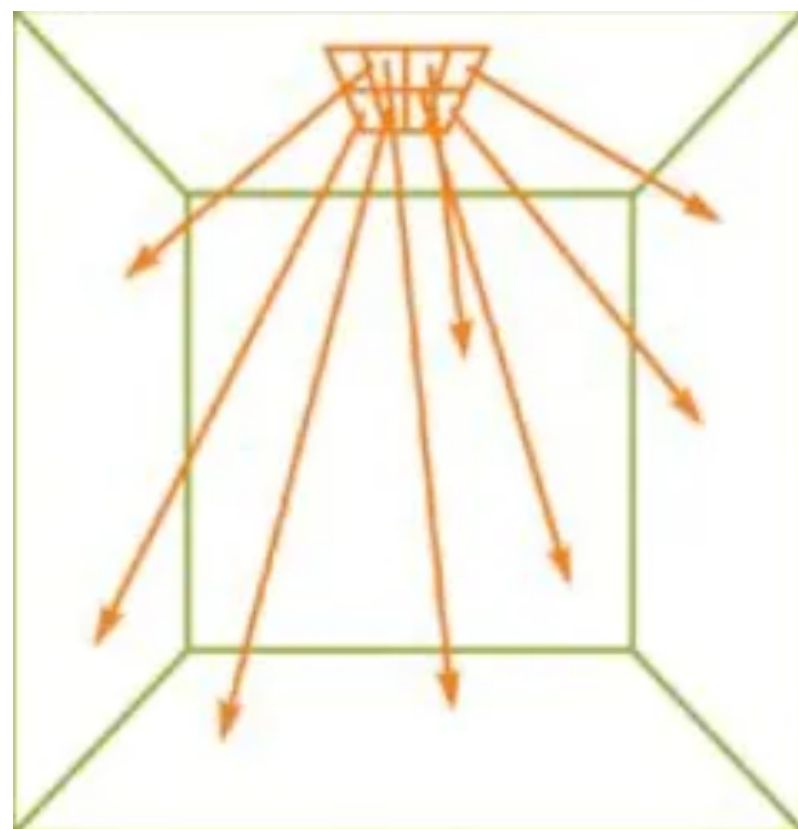
# Introduzione

- Nell'ambito della grafica 3D, l'argoritmo di Radiosità (Radiosity) è una tecnica avanzata di Illuminazione Globale, view-independent, per il rendering di componenti diffuse.
- Promessa: La componente di luce diffusa rimbalza su una superficie nella scena, indipendentemente dalla fonte, ed essa interagisce con altre superfici.
- Lo scopo dell'algoritmo è determinare quanta "energia" viene emessa dalla superficie in rifrazione.



# Proprietà e funzionamento

- Non esiste distinzione fra un oggetto normale e una luce propria.
- Ogni superficie nella scena viene divisa in **Patches**.
- La Radiosity si occupa di determinare quanta energia trasmessa dalla patch A alla patch B.



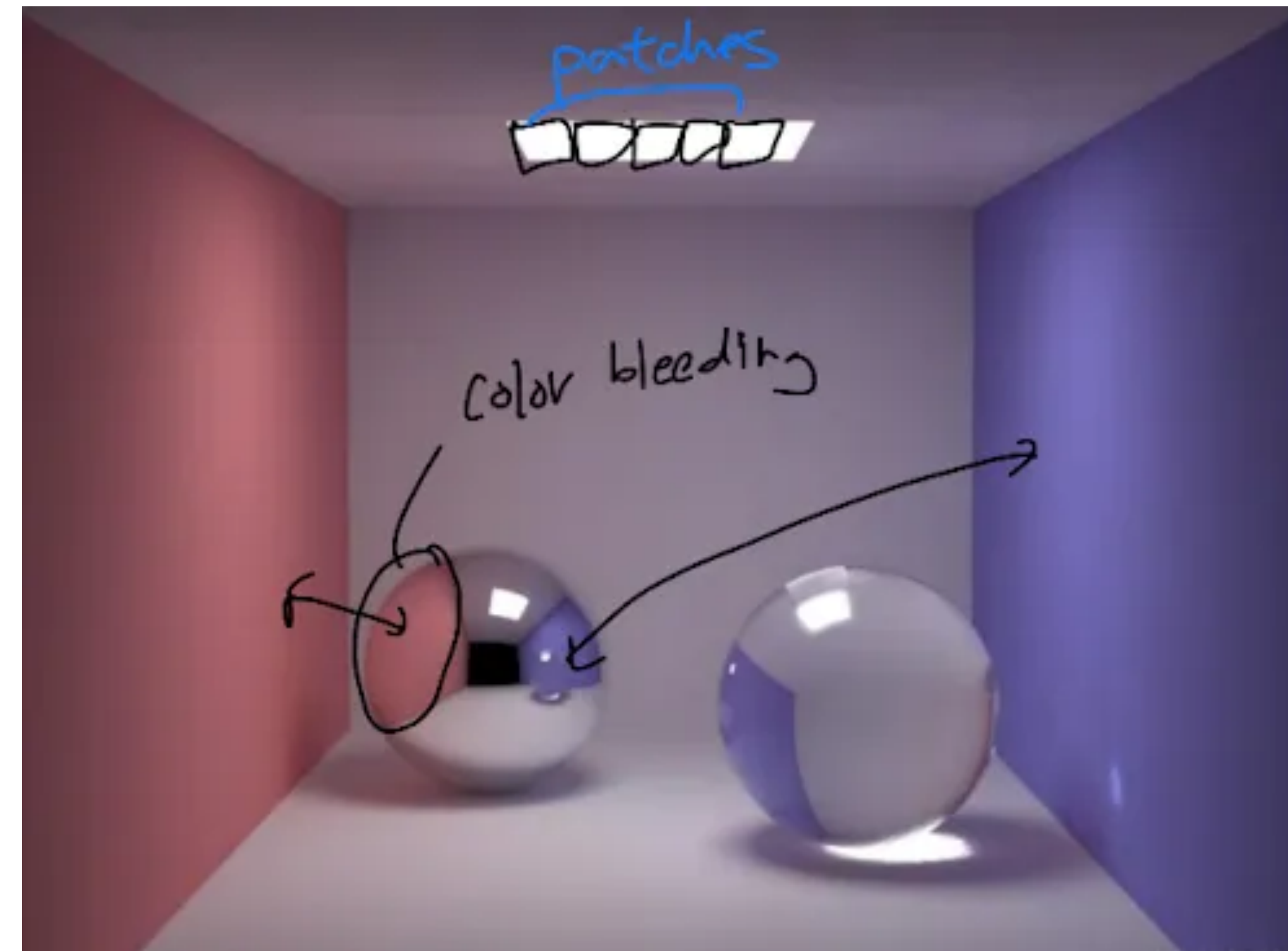
*Esempio visuale di una scena divisa in patches*

<https://medium.com/@coderfromnineteen/computer-graphics-theory-radiosity-global-illumination-8c473607e6cd>



# Proprietà e funzionamento

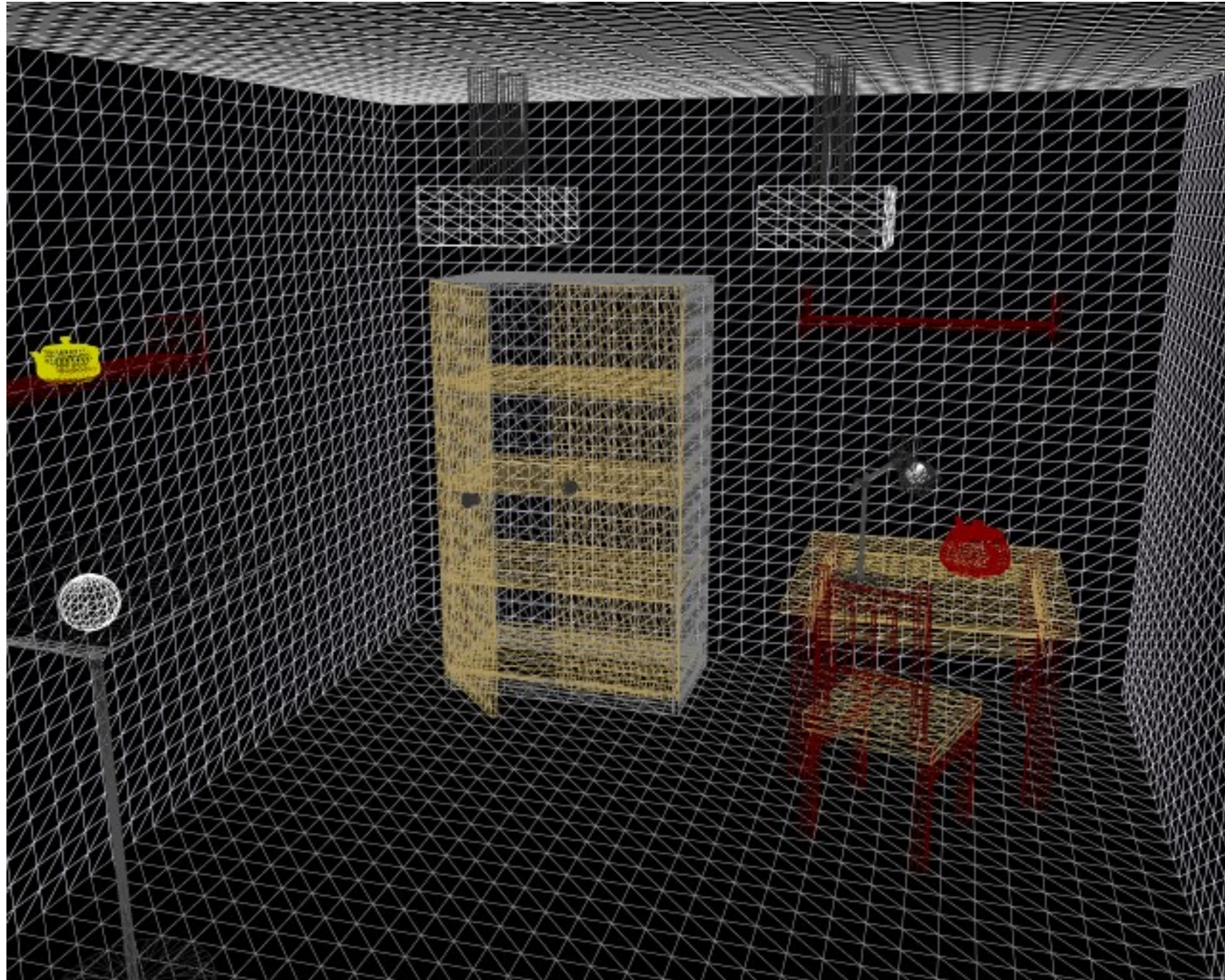
- Questo processo porta a una serie di proprietà interessanti:
  - Color Bleeding: rimbalzando su una superficie 1, una superficie 2 assume sfumature di colore dalla prima (in base alla riflessibilità del secondo oggetto)
  - Soft Shadows: antialiasing sulle ombre proiettare.



*Esempio di scena renderizzata con Radiosity*  
<https://medium.com/@coderfromnineteen/computer-graphics-theory-radiosity-global-illumination-8c473607e6cd>



# Proprietà e funzionamento



*Patch division e varie iterazioni dell'algoritmo di Radiosity.*  
<https://dudka.cz/rvv>



# L'Algoritmo

- Consideriamo ogni elemento come una sorgente di luce che, oltre ad emettere la propria luce, diffonde quella proveniente dalle altre sorgenti (ma non la riflette specularmente).
- L'equazione della Radiosity discreta B per la patch i risulta:

$$B_i = E_i + \rho_i \sum_{j=1}^n F_{ij} B_j$$

Dove:

- $E_i$  è la costante di emissività
- $\rho_i$  è la costante di riflessività
- $\sum_{j=1}^n F_{ij} B_j$  è la sommatoria della luce che incide sulla patch i da tutte le altre n patch..

# L'Algoritmo

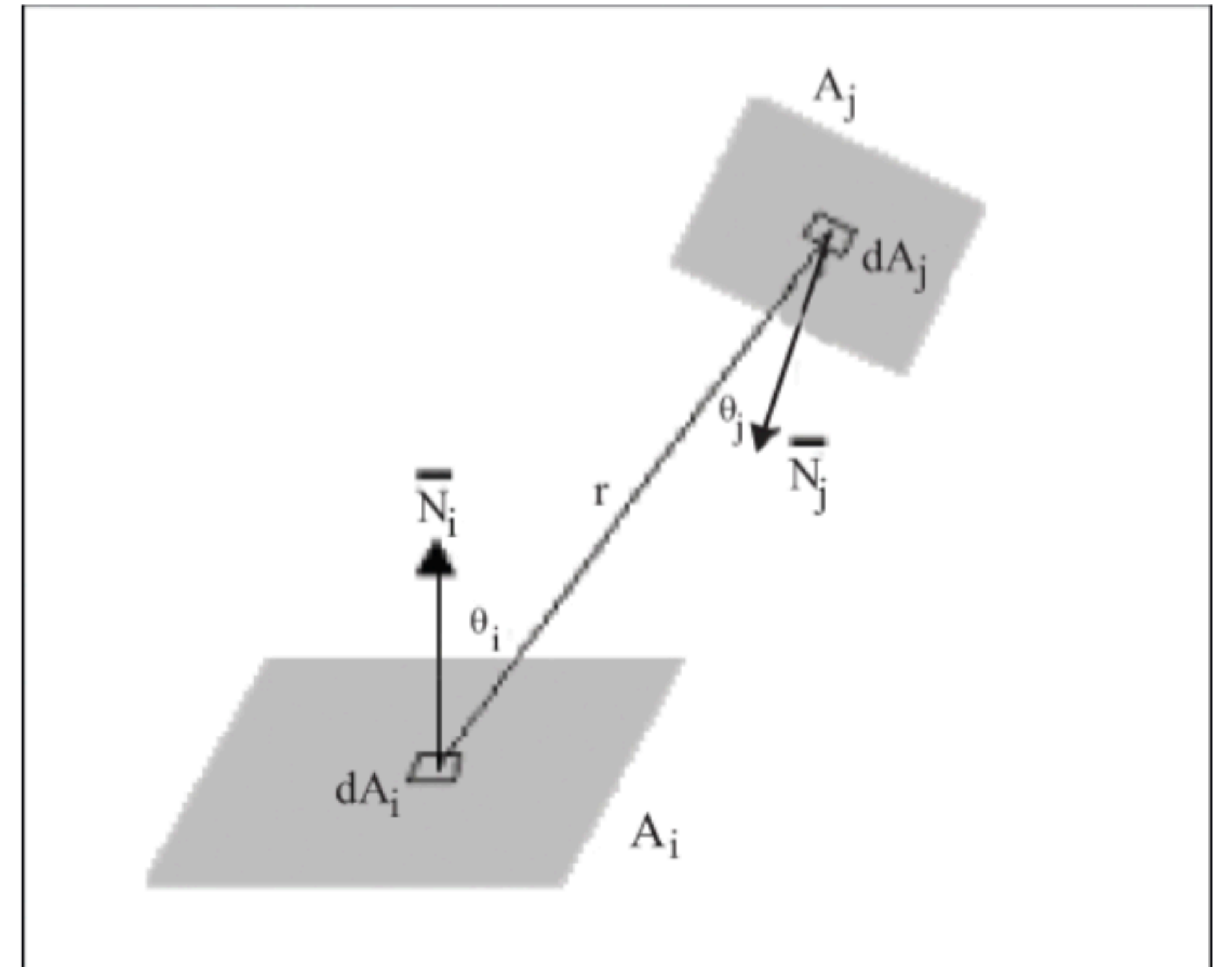
- L'algoritmo presentato presenta una Complessità di  $O(n^2)$ .
- Per ovviare a ciò, si è sviluppato un metodo per la risoluzione detto Progressive Refinement Radiosity (PRR):
  - Agisce attraverso varie interazione, non calcolando tutto assieme.
  - Viene scelta una patch  $j$  (lo "shooter") e usa la sua Radiosity attuale per aggiornare le altre. Accumula gradualmente i risultati, perfezionando la soluzione.
  - Complessità di  $O(n*s)$ , dove  $s$  sono il numero di iterazioni.

# Form Factor

- Il Form Factor ( $F_{ij}$ ): coefficiente che rappresenta matematicamente quanto bene la luce che proviene dalla Patch A raggiunga la Patch B. Influenzato non solo dalle proprietà delle patch stesse ma anche dalla scena stessa.
- Quanto bene A e B riescono a “vedersi”?
- Come si calcola il  $F_{ij}$ ?
  - Calcolo geometrico con media integrale

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} H_{ij} dA_j dA_i$$

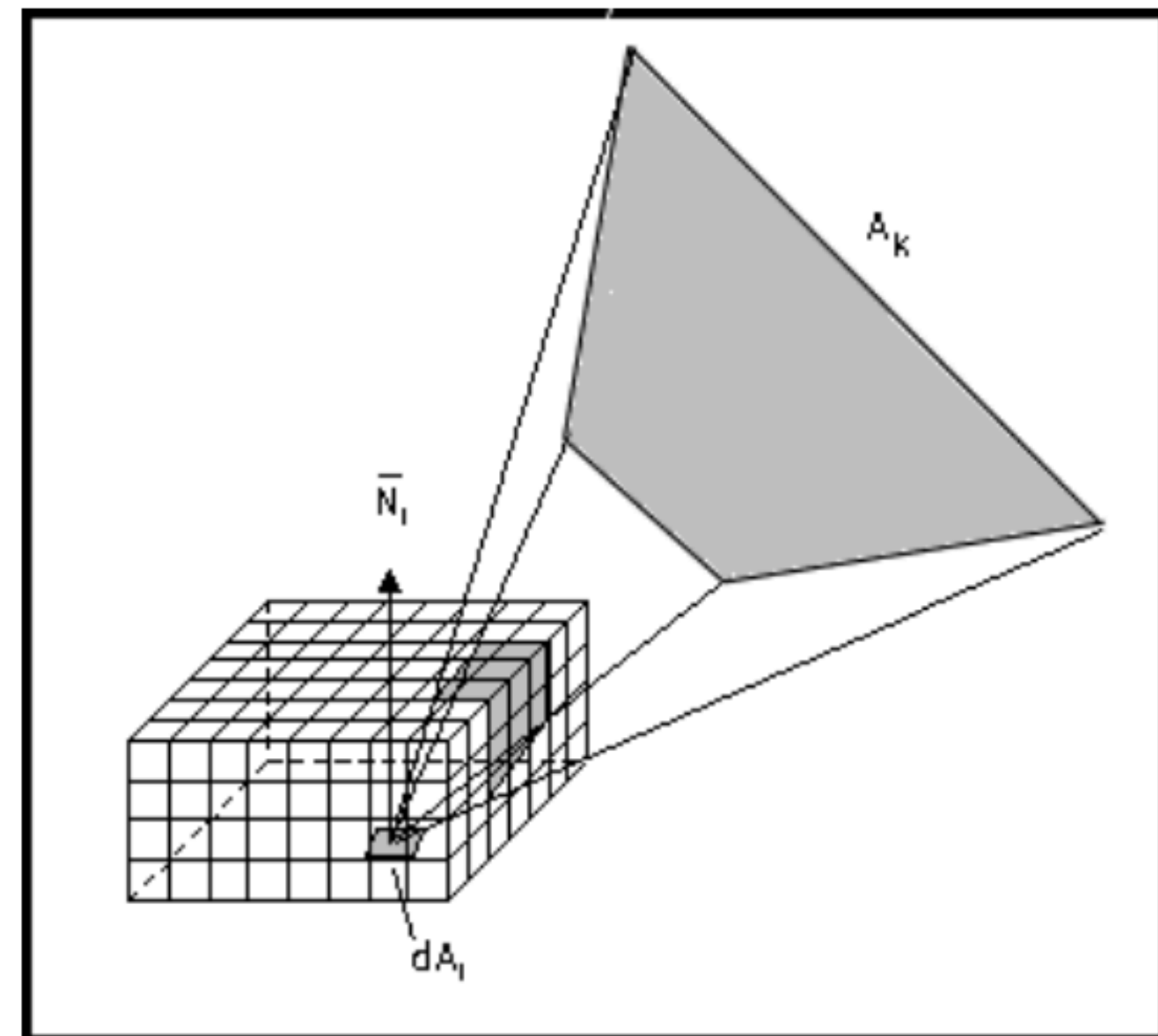
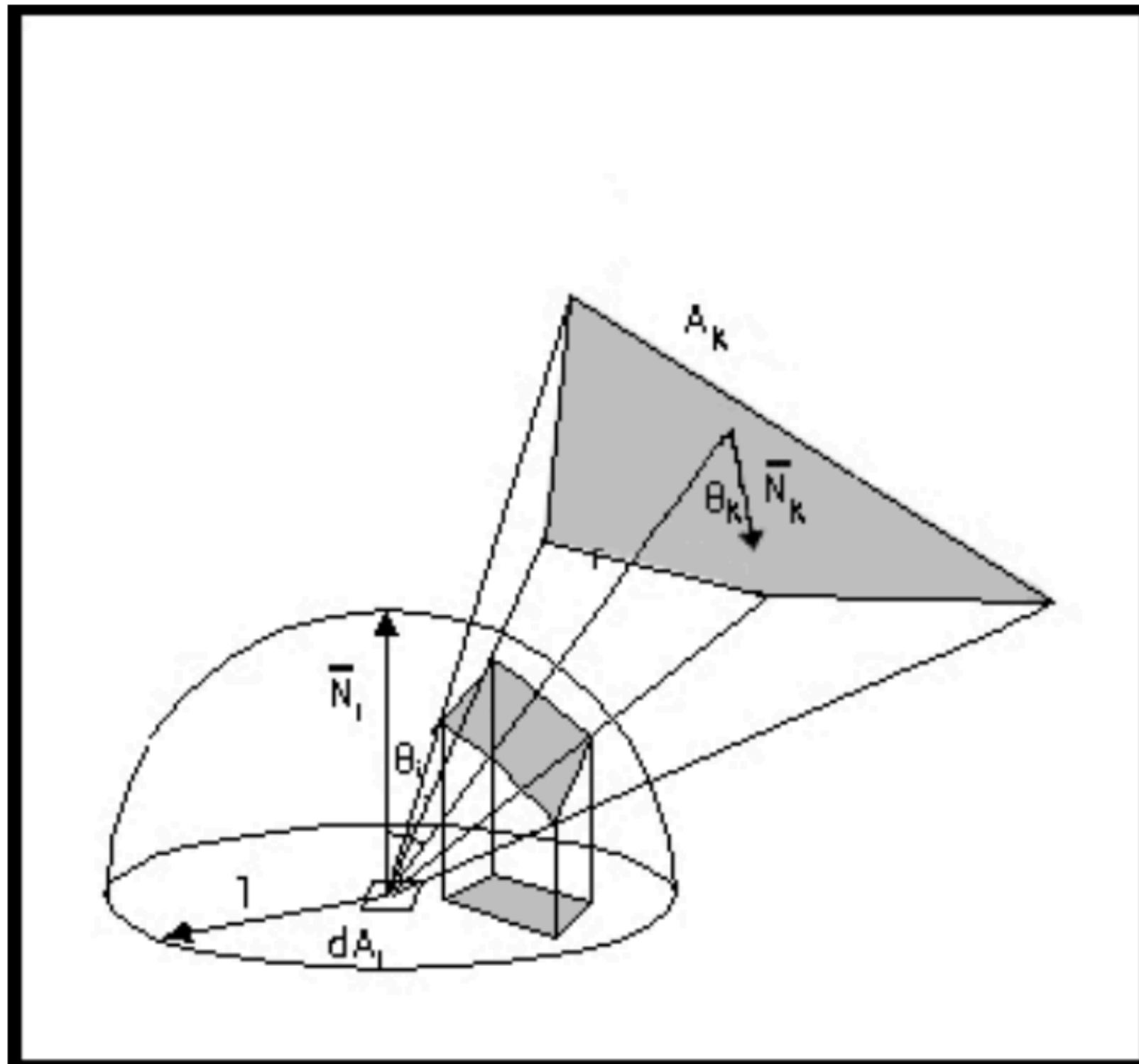
*Introduzione alla radiosità,  
Dipartimento di Matematica dell'Università degli studi di  
Roma "Tor Vergata"*



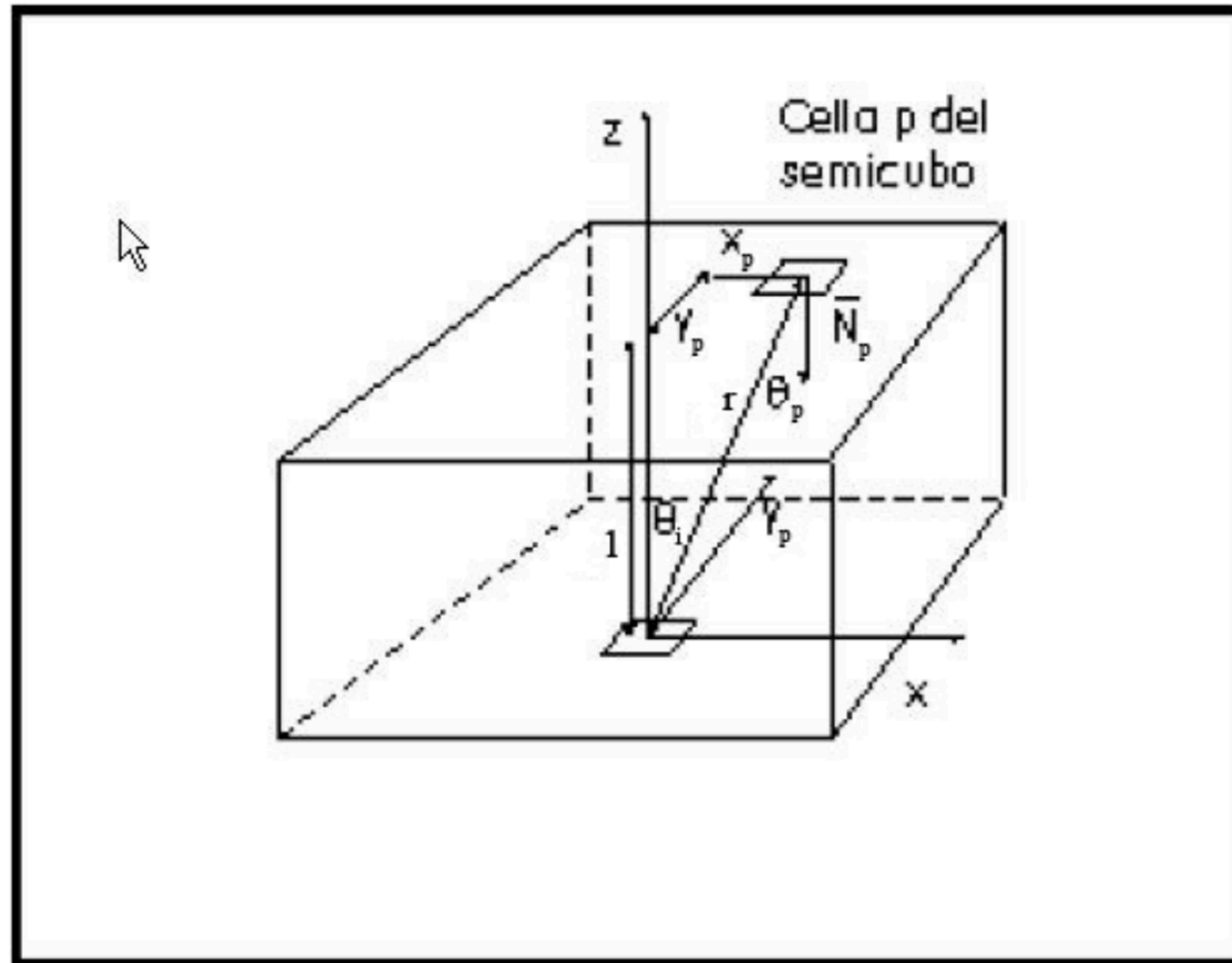


# Form Factor

- Algoritmo non usabile, troppo oneroso da calcolare.
- Si ricorre a delle approssimazioni geometriche: proiezione su emisfera o proiezione su un emicubo. Se poniamo un emicubo (mezzo cubo) sopra una patch  $i$ , possiamo usare un rendering z-buffer per stimare quanto di ogni altra patch è “visibile” da  $i$ . Analizziamo la scena completa fin dall’inizio.



# Form Factor



$$\Delta F_p = \frac{\cos \theta_i \cos \theta_p}{\pi r^2} \Delta A$$

Dove:

- $\theta_p$  è l'angolo tra la normale alla cella p e il vettore r che congiunge il centro di  $dA_i$  (cioè il centro del cubo) col centro di p.
- $\theta_i$  è l'angolo di incidenza.
- $\Delta A$  è l'area della cella.

Riflessione su una patch soprastante: 
$$\Delta F_p = \frac{1}{\pi(1 + x_p^2 + y_p^2)^2} \Delta A$$

Riflessione su una patch laterale: 
$$\Delta F_p = \frac{z_p}{\pi(1 + x_p^2 + y_p^2)^2} \Delta A$$

Va infine sommato il contributo di tutte le patch N per ottenere il form factor complessivo.

$$F_{ij} = \sum_{n=1}^N \Delta F_n$$

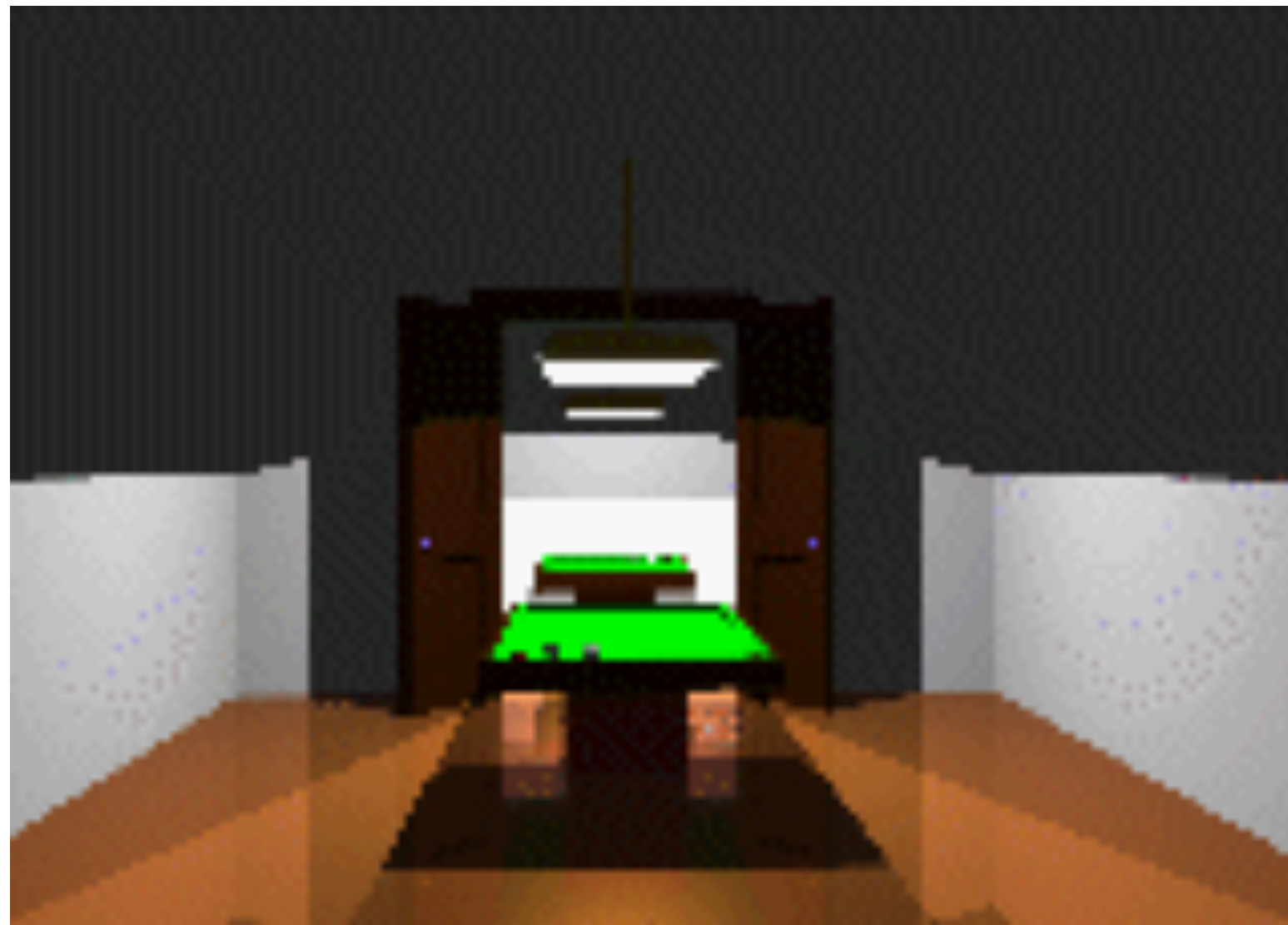


# Confronto ed interazioni con il Ray-tracing

- Questi 2 algoritmi a prima vista possono sembrare simili, potendo essere entrambi usati per la rappresentazione dell'illuminazione globale all'interno di una scena.
- Differenze:
  - Il Ray-Tracing si concentra sulla rappresentazione di riflessi e del calcolo di componenti speculari (riflessi, rifrazioni, ombre dure) ignorando le componenti diffuse, Radiosity considera solo le componenti diffuse.
  - Il Ray-Tracing è view-dependent, Radiosity è view-independent.

# Confronto ed interazioni con il Ray-tracing

- Queste 2 tecniche posso essere combinate in una singola scena.



*Sinistra: Scena con solo Ray-Tracing attivo*

*Centro: Scena con solo la Radiosity attiva*

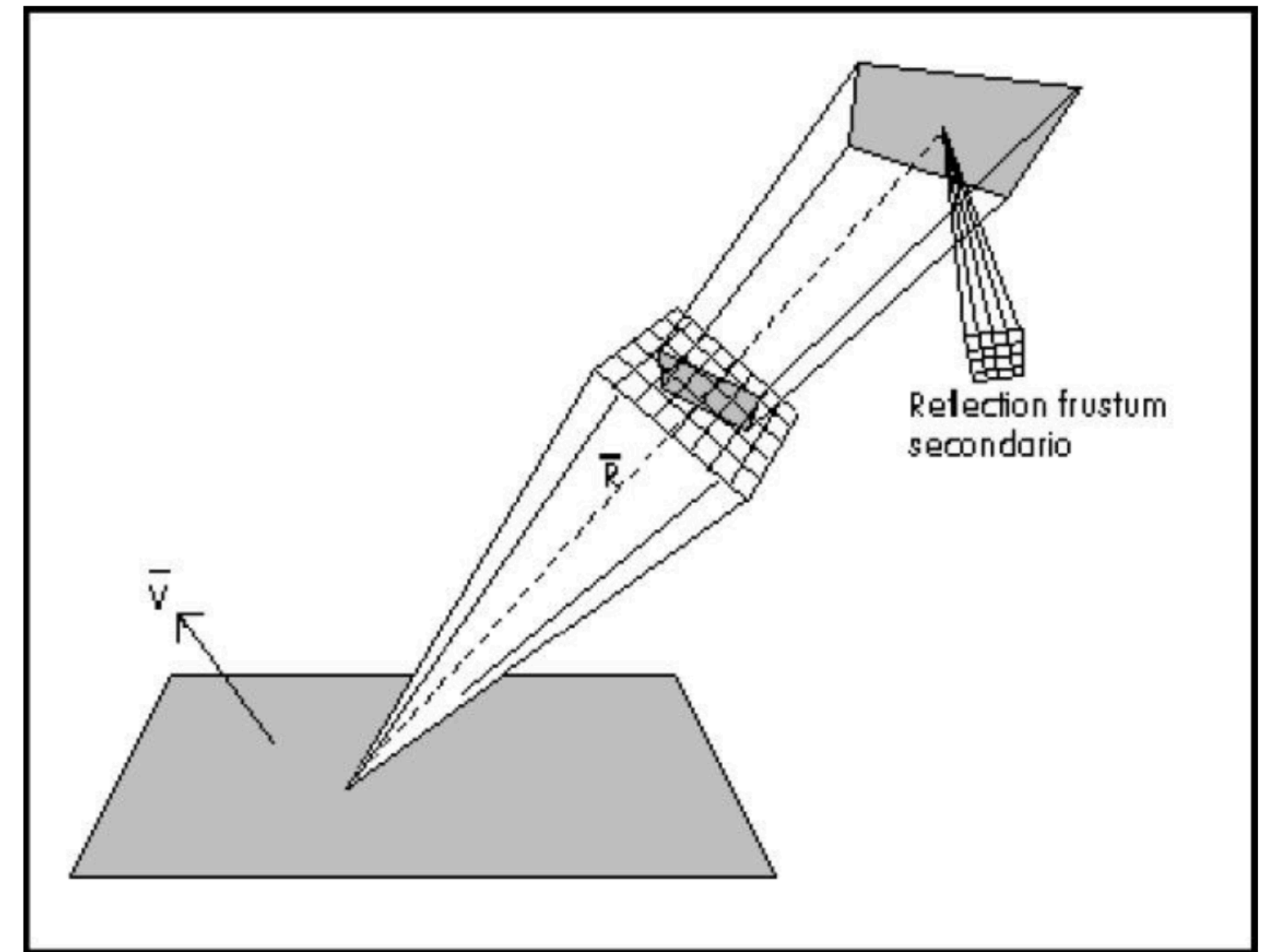
*Destra: Scena col metodo a 2 passi*

*<https://www.cg.tuwien.ac.at/research/rendering/rays-radio/>*



# Confronto ed interazioni con il Ray-tracing

- Non si può semplicemente combinare i 2 metodi assieme.
- Approccio a 2 passi:
  - Radiosity ampliata
  - Ray Tracing Ampliato
- Purtroppo, come è possibile immaginare, entrambi i passi risultano molto onerosi.



*Introduzione alla radiosità,  
Dipartimento di Matematica dell'Università degli studi di  
Roma "Tor Vergata"*